

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平11-272623

(43)公開日 平成11年(1999)10月8日

(51)Int.Cl.<sup>8</sup>

G 0 6 F 15/16  
9/46  
13/00

識別記号

3 8 0  
3 6 0  
3 5 7

F I

G 0 6 F 15/16  
9/46  
13/00

3 8 0 Z  
3 6 0 C  
3 5 7 Z

審査請求 未請求 請求項の数7 O L (全 15 頁)

(21)出願番号

特願平10-70493

(22)出願日

平成10年(1998)3月19日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番  
1号

(72)発明者 熊本 乃親

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(72)発明者 田中 竜太

神奈川県川崎市中原区上小田中4丁目1番  
1号 富士通株式会社内

(74)代理人 弁理士 今村 辰夫 (外1名)

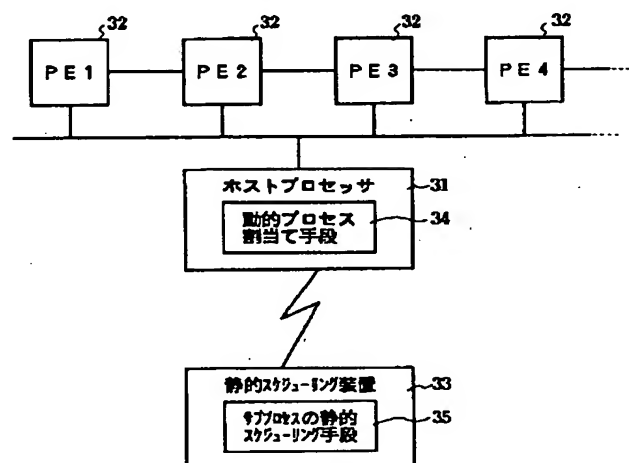
(54)【発明の名称】 プロセス割当て方法、静的スケジューリング装置、及び記録媒体

(57)【要約】

【課題】本発明は、プロセス割り当て方法、静的プロセス割り当て装置及び記録媒体に関し、リニアアレイ型プロセッサシステムにおける動的プロセス割当てによってシステム全体の能力を最大限引き出すために、サブプロセスの実行スケジュールを、予め静的スケジューリング手段によって求めておくことで、効率的な処理を可能にする。

【解決手段】複数のプロセッサ32を直列接続したリニアアレイ型マルチプロセッサシステムの各プロセッサ32に対し、例えば、ホストプロセッサ31の動的プロセス割当て手段34が複数のサブプロセスを構成要素として含むプロセスを動的に割当てする時、静的スケジューリング装置33に設けたサブプロセスの静的スケジューリング手段35が、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておくようにした。

## システムの説明図



Best Available Copy

1

## 【特許請求の範囲】

【請求項 1】複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるプロセス割当て方法において、前記動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、サブプロセスの静的スケジュールリング手段により静的に求めておくことを特徴としたプロセス割当て方法。

【請求項 2】前記サブプロセスの静的スケジュールリング手段が、そのプロセスが要求するプロセッサ数最小となるスケジュール情報を得ることを特徴とした請求項 1 記載のプロセス割当て方法。

【請求項 3】前記サブプロセスの静的スケジュールリング手段が、そのプロセス単体の処理時間最小となるスケジュール情報を得ることを特徴とした請求項 1 記載のプロセス割当て方法。

【請求項 4】前記サブプロセスの静的スケジュールリング手段が、そのプロセスが要求するプロセッサ間データ転送量最小となるスケジュール情報を得ることを特徴とした請求項 1 記載のプロセス割当て方法。

【請求項 5】前記サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュール情報を選択して割当てることを特徴とした請求項 1 記載のプロセス割当て方法。

【請求項 6】複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるためのサブプロセスのスケジュール情報を、予め、静的に求めておくサブプロセスの静的スケジュールリング手段を備えていることを特徴とした静的スケジュールリング装置。

【請求項 7】コンピュータに、複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるためのサブプロセスのスケジュール情報を、予め、静的に求めておく手順を実行させるためのプログラムを記録したコンピュータ読取可能な記録媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるプロセス割当て方法、そのための静的スケジュールリング装置、及び記録媒体に関する。

【0002】近年、デジタル通信システム、マルチメディアシステムなど、実時間処理が要求される分野においては、高速、低コストで、かつ複雑なアルゴリズムに

2

柔軟に対応可能なデジタル信号処理プロセッサ(DSP)を始めとして、信号処理を高速に実行可能なプロセッサが広く使われている。

【0003】ところが、画像処理、音響処理CG、などの処理を複数同時に処理しようとした場合、単一のプロセッサでは実時間で処理を行うことが困難となっている。また、単一プロセッサで実現される処理においても、処理すべきデータ量が増加した場合や、要求される速度が早くなった場合、単一のプロセッサでは実現できなくなってしまうことがある。そこで、複数のプロセッサを使用して処理の高速化を図る必要がある。

## 【0004】

【従来の技術】以下、図に基づいて従来例を説明する。

## §1：従来例1の説明

従来、マルチプロセッサシステムにおいて、各プロセッサにプロセスを割当てる場合、空いているプロセッサから順番にプロセスを割当てる方法がとられている。これは、リニアアレイ型マルチプロセッサシステムではプロセッサ間通信を行う場合、転送経路が1つ(直線状)又は2つ(リング状)しかないため、通信時間をほとんど考慮する必要がないためである。

【0005】また、このように空いているプロセッサを見つけて順に割当てる方法では、動的なプロセス割当てに際し、重要な割当てに要する時間が小さいという利点がある。しかしながら、アプリケーションによっては、プロセッサ間のデータ転送に用いるリンクに流れるデータ転送量がかなり多い場合があり、このような場合、或る特定のプロセッサ間のリンクのデータ転送量が大きくなり、他のプロセッサ間のリンクのデータ転送流が小さくなってしまうことが起こる。

【0006】このような場合、プロセッサの能力はまだ十分に空いているにも関わらず、データ転送ができないため、全体の処理が実時間で実現不可能になってしまうことがある。

## 【0007】§2：従来例2の説明

データ転送量を考慮した割当て方法としては、例えば、特開平9-34847号公報に記載された発明がある。以下、この例を従来例2として説明する。従来例2では、通信路にスイッチを置き、そのスイッチでデータ転送量を計数し、それに応じて通信経路を変更することで、データ転送量を考慮したプロセスの割当てを行う。

【0008】ところが、このプロセス割当て方法では、通信経路が複数あれば効果はあるが、リニアアレイ型プロセッサシステムでは、通信路は1つ、リンク構成でも2つしかなく、ほとんど効果がない。

## 【0009】§3：従来例3の説明

プロセッサ間のデータ転送量の最大値が最小となるプロセスを割当てることで、プロセッサの能力を生かした処理の実現を可能にしたものが既に提案されている。ところが、ここで割当てる各プロセスは、一般に複数のプロ

50

3

セッサ資源を要求するため、単一プロセッサで処理されるサブプロセスを複数接続した構成になっている。

【0010】従って、このサブプロセスの実行スケジュールによって、そのプロセスが要求する資源は変化する。このようなサブプロセスの実行スケジュールについては、特開平 8-1 5 2 9 0 3 号公報に示されているように、予め制御グループを設定し、グループ内をスケジューリングする方法が知られている。

【0011】しかし、この方法では、制御グループを先に静的に決定してしまうため、グループ間のデータ転送量の負荷分散を行うには、その都度、制御グループの割当て変更が必要となり、負荷分散が動的に行われない。

【0012】§ 4：従来例 4 の説明

複数プロセッサで構成されるシステムの中で、複数のプロセッサを直列接続して得られるリニアアレイ型マルチプロセッサシステムが、先に提出した特願平 9-2 2 1 6 1 7 号（以下「従来例 4」と記す）に記載されている。この従来例 4 に示されているリニアアレイ型マルチプロセッサシステムは、構成が簡単で、最も安価なシステムを構成できる。

【0013】前記リニアアレイ型マルチプロセッサシステムでは、プロセッサは左右のいずれかのプロセッサへは、直接データ転送できるが、それ以外のプロセッサへのデータ転送は、他のプロセッサを経由してしかデータ転送できない。従って、複数のプロセッサに対し、ユーザの要求によりダイナミックに処理を行ったり、停止する場合、プロセスの割当て方によって、複数のプロセッサの能力を十分に生かしきれない。

【0014】以下、従来例 4 について、図を参照しながらその概要を説明する。図 8 は従来例の説明図（その 1）であり、信号処理アクセラレータ（リニアアレイ型プロセッサシステム）の説明図である。以下、図 8 に基づいて、信号処理アクセラレータ（リニアアレイ型プロセッサシステム）の構成を説明する。

【0015】図 8 に示した信号処理アクセラレータは、複数で同一の情報処理ユニット 10 を含む。各情報処理ユニット 10 は互いに接続されると共に、ホストメモリバス 30 に接続される。各情報処理ユニット 10 は、信号処理プロセッサ 11、命令キャッシュ 12、データ RAM 13、リンク制御部 14、15、メインキャッシュ 16、リンクキャッシュ 17、DRAM コントローラ 19 を含む。なお、前記信号処理プロセッサ 11 及びデータ RAM 13 は、信号処理部 25 を構成する。また、リンク制御部 14、15、メインキャッシュ 16、及びリンクキャッシュ 17 は、通信制御部 26 を構成する。リンク制御部 14、15 には通信リンク 20 が接続される。各情報処理ユニット 10 は通信リンク 20 を介して直列に配列され、隣接する情報処理ユニット 10 と通信する。通信内容を或る情報処理ユニット 10 から次の情報処理ユニット 10 へと伝達していくことで、任意の情

4

報処理ユニット間で通信を行うことができる。この場合、図 8 では、3 個の情報処理ユニット 10 が示されているが、個数は 3 個に限らず任意である。

【0016】また、各情報処理ユニット 10 は DRAM コントローラ 19 を介してホストメモリバス 30 に接続される。更に、ホストプロセッサ 31 がホストメモリバス 30 に接続される。信号処理プロセッサ 11 は信号処理機能を実現するプロセッサであり、命令キャッシュ 12 は信号処理プロセッサ 11 が頻繁に用いる命令を格納しておくためのキャッシュメモリである。

【0017】なお、信号処理プロセッサ 11 が実行するプログラムは、命令キャッシュ 12 以外に DRAM 18 に格納されている。データ RAM 13 は、信号処理プロセッサ 11 がデータ処理する際に中間結果を格納するため等のワーク領域として用いられる。メインキャッシュ 16 及びリンクキャッシュ 17 は、信号処理プロセッサ 11 が処理するデータを格納しておくためのキャッシュメモリである。

【0018】メインキャッシュ 16 には、情報処理ユニット 10 自身の DRAM 18 から取り込んだデータを格納し、リンクキャッシュ 17 には、リンク制御部 14、15 を介して他の情報処理ユニット 10 から取り込んだデータを格納する。

【0019】メインキャッシュ 16 のデータがスワップアウトされても、そのデータが必要になった場合には、自分の DRAM 18 からデータを再度読み出すことができる。それに対して、リンクキャッシュ 17 のデータがスワップアウトされると、別の情報処理ユニット 10 から通信リンク 20 を介してデータを再度持ってくる必要がある。

【0020】メインキャッシュ 16 とリンクキャッシュ 17 を同一のキャッシュメモリとしてしまうと、例えば、通信負荷が重い状態なのに、自身の DRAM 18 からのデータをキャッシュメモリに格納することによって、他の情報処理ユニット 10 から獲得したデータをスワップアウトしてしまう等の問題が生じる。この例では、メインキャッシュ 16 とリンクキャッシュ 17 とを機能別に分けている。

【0021】データ処理を行う際、複数の情報処理ユニット 10 は互いに通信しながら、並列処理、或いはパイプライン処理を行う。たとえば、いくつかの情報処理ユニット 10 が画像データ処理を並列に実行している間に、別の幾つかの情報処理ユニット 10 が音声データを並列に処理すること等が可能である。

【0022】複数の情報処理ユニット 10 間の通信は、通信リンク 20 によって行われる。従って、ホストメモリバス 30 は、複数の情報処理ユニット 10 間での通信には全く関与せず、ホストプロセッサ 31 が実行する OS プロセス等の他のプロセスにデータ転送経路を提供できる。

5

【0023】各情報処理ユニット10は処理後のデータをDRAM18に書き込む。ホストプロセッサ31はホストメモリバス30を介してDRAM18にアクセスすることによって、データ処理後のデータを書き込むことができる。図8の信号処理アクセラレータは、ホストメモリバス30を介在しないで通信可能な複数の情報処理ユニット10を設けて、それら情報処理ユニット10に並列処理を実行させるので、バス競合によりデータ処理速度が低下することなく、高速な信号処理を実現することができる。

【0024】また、画像処理プロセスや音声処理プロセス等の複数のプロセスの各々に、別の情報処理ユニット10を割当てることができるので、複数の異なった信号を処理する必要があるマルチメディア信号処理に適している。また、信号処理部25（信号処理プロセッサ11、命令キャッシュ12、データRAM13）、通信制御部26（キャッシュメモリ16、及び17と、リンク制御部14及び15）、及びメモリ（DRAM18及びDRAMコントローラ19）を1つのチップ上に集積回路化して、従来のメモリと同等な形でパーソナルコンピュータに搭載することができる。

【0025】従って、コストを従来のメモリバスに重複させることが可能であると共に、メモリバス内に埋め込まれた信号処理アクセラレータをソフトウェアで活用できるという利点がある。従って、ハードウェアの追加、拡張にかかる費用を削減すると共に、機能拡張に優れたシステムを構築できる。

【0026】§5：プロセス割当て方法の説明・・・図9参照

図9は従来例の説明図（その2）であり、2つの異なったプロセス割当ての方法（A図は例1、B図は例2）を示した図である。以下、図9に基づいて、2つの異なったプロセス割当ての方法を説明する。図9のA図のように、プロセス1をプロセッサエレメントPE1及びPE3に割当て、プロセス2をプロセッサエレメントPE2及びPE4に割当てた場合を説明する。

【0027】一つのプロセッサ内で2つのPE間のデータ転送量をMとすれば、PE1及びPE3間でPE2を介してのデータ転送量はMとなり、また、PE2及びPE4間でPE3を介してのデータ転送量もMとなる。従って、データ転送量は、PE1及びPE2間でM、PE2及びPE3間で2M、PE3及びPE4間でMとなる。

【0028】次に、図9のB図に示したように、プロセス1をプロセッサエレメントPE1及びPE2に割当て、プロセス2をプロセッサエレメントPE3及びPE4に割当てた場合を説明する。この場合、PE1及びPE2の間でのデータ転送量がM、PE3及びPE4間でのデータ転送量がMとなり、PE2及びPE3間ではデータ転送は行われない。

6

【0029】各PE間のデータ転送能力が、たとえば、1.5Mであるとする、図9のA図の場合（例1）には、一方のプロセスの処理が実現不可能となるのに対して、図9のB図の場合（例2）には、両方のプロセスの処理を同時に実現できる。このようにプロセスの割当て方によっては各リンクのデータ転送量が異なる結果となり、複数のプロセスの完全な同時処理が可能な場合と、不可能な場合とが生じる。

【0030】同時処理が不可能な場合には、当然ながら全体でのデータ処理速度が劣化することになる。しかも、他のプロセッサエレメントから要求される数やタイミングは、実際の要求が発行される前には全く未知であるため、プロセスの割当てを動的に行う必要がある。従って、効率の良い動的プロセス割当てアルゴリズムが必要になる。

【0031】§6：動的プロセス割当てアルゴリズムの説明・・・図10～13参照

図10～図13は従来例の説明図（その3）～（その6）であり、動的プロセス割当てアルゴリズムを示した図である。以下、図10～図13に基づいて、動的プロセス割当てアルゴリズムを説明する。

【0032】(1)：メインルーチンの説明・・・図10参照

図10は従来例の説明図（その3）であり、動的プロセス割当てアルゴリズムのメインルーチンを示すフローチャートである。以下、図10に基づいて、動的プロセス割当てアルゴリズムのメインルーチンを説明する。なお、S1～S5は各処理ステップを示す。

【0033】この動的プロセス割当てアルゴリズムは、リソース管理プログラム（図示省略）が実行する処理である。この動的プロセス割当てアルゴリズムは、以下の2つの指標に基づいてリソース（この場合のリソースはPEを指す）の割当てを行う。第1の指標は、割当てるプロセスのデータ転送ができるだけ他のデータ転送と重ならないことである。第2の指標は、当該プロセスを割当てた結果、次のプロセスを割当ての際に、できるだけ他のデータ転送と重ならないことである。

【0034】先ず、第1の指標によって、割当てようとしているプロセスによって生じる通信リンク20上のデータ転送量の最大値が最小になるように設定する。次に、同一の最大値を有する割当て方の中から、第2の指標を用いて、次に割当てるプロセスができるだけ阻害されないような割当て方を求める。

【0035】図10に示したように、このアルゴリズムにおいては、PEを一個要求する場合と、複数個要求する場合とで、別々に割当て方を求める。これはPEを1個だけ要求する場合には、当該プロセスによる通信リンク20上のデータ転送は生じないために、次のプロセス割当てに対する影響を考えればよいからである。

【0036】それに対してPEを複数個割当てる場合に

は、通信リンク 20 を介したデータ転送が必要であるため、そのプロセスの割当て方によって、当該プロセス自身の効率が影響を受けてしまう。

【0037】図 10 のステップ S1 において、フリーリソースとして利用可能な PE の数を検査する。利用可能な PE が存在しない場合にはアルゴリズムを終了し、存在する場合にはステップ S2 に進む。ステップ S2 において、要求される PE の数が 1 個であるのか否かを判断する。1 個の場合はステップ S3 に進み、複数の場合はステップ S4 に進む。

【0038】ステップ S3 において、PE の 1 個要求に対する割当てを行う。割当てが失敗した場合にはアルゴリズムを終了し、成功した場合は次にステップ S5 に進む。ステップ S4 において、PE の複数個要求に対する割当てを行う。割当てが失敗した場合にはアルゴリズムを終了し、成功した場合にはステップ S5 に進む。ステップ S5 において、プロセス ID を更新する。すなわち、新規のプロセスに、新規のプロセス ID を割当てる。これでアルゴリズムを終了する。

【0039】(2) : PE の 1 個要求に対する割当ての説明・・・図 11 参照

図 11 は、従来例の説明図 (その 4) であり、図 10 のステップ S3 における PE の 1 個要求に対する割当て処理を示すフローチャートである。なお、S11～S17 は各処理ステップを示す。

【0040】ステップ S11 において、利用可能な PE を検索する。次に、ステップ S12 において、全ての利用可能な PE に対してループを構成する。すなわち、全ての利用可能な PE に対して以下のステップを繰り返す。ステップ S13 において、1 個の PE を仮割当てする。ステップ S14 において、次の割当て効率を計算する。なお、割当て効率を計算した結果を「Result」と記す。

【0041】ステップ S15 において、Result が最小の値を保持する。ステップ S16 においてループを終了する。ステップ S17 において、Result が最小の値である PE を、実際にプロセスに割当てる。以上で処理を終了する。

【0042】(3) : PE の複数個要求に対する割当ての説明・・・図 12 参照

図 12 は従来例の説明図 (その 5) であり、PE の複数個要求に対する割当てを示すフローチャートである。以下、図 10 のステップ S4 における PE の複数個要求に対する割当てを説明する。なお、S21～S31 は各処理ステップを示す。

【0043】ステップ S21 において、利用可能な PE を検索する。ステップ S22 において、要求される個数の利用可能な PE の全ての組み合わせに対して、第 1 ループを構成する。すなわち、要求される個数の利用可能な PE の全ての組み合わせに対して、以下のステップを

繰り返す。

【0044】ステップ S23 において、当該プロセスの割当てによって生じる通信リンク 20 のデータ転送量を計算する。ステップ S24 において、通信リンク 20 のデータ転送量の最大値が最小の割当てを保持する。ステップ S25 において、第 1 のループを終了する。ステップ S26 において、通信リンク 20 のデータ転送量の増加が最小であるような割当てを第 1 のループで選択された割当てとして、全ての選択された割当てに対して第 2

10

のループを構成する。

【0045】ステップ S27 において、選択された割当て方で、複数個の PE を仮割当てする。ステップ S28 において、次の割当て効率を計算する。なお、次の割当て効率を計算した結果を、Result と記す。ステップ S29 において、結果 Result が最小の割当てを保持する。ステップ S30 において、第 2 のループを終了する。ステップ S31 において、Result が最小の割当てである割当て方で、実際にプロセスに割当てる。以上の処理を終了する。

20

【0046】(4) : 次の割当て効率計算処理の説明・・・図 13 参照

図 13 は従来例の説明図 (その 6) であり、図 11 のステップ S14 及び図 12 のステップ S28 に於ける次の割当て効率計算処理を示すフローチャートである。以下、図 13 に基づいて、前述した次の割当て効率計算処理を説明する。なお、S41～S43 は各処理ステップを示す。

【0047】ステップ S41 において、利用可能な PE の内で、最も左端の PE を選択して PE-L とする。ステップ S42 において、利用可能な PE の内でも最も右端の PE を選択して PE-R とする。ステップ S43 において、PE-L から PE-R までの通信リンク数を求め、この通信リンク数を Result とする。以上で処理を終了する。

30

【0048】図 13 のフローチャートにおいては、最も左端の PE と最も右端の PE を選択して、両 PE 間での通信リンク数を求める。すなわち、この通信リンク数によって、次のプロセスを割当てする際の割当て効率を求めていることになる。これは以下のように考えれば良い。すなわち、両 PE 間での通信リンク数が少ないということは、利用可能な PE がまとまって存在していることを意味する。逆に、両 PE 間での通信リンク数が多いと、利用可能な PE が広い範囲に広がって存在していることになる。

40

【0049】狭い範囲にまとまった PE にプロセスを割当てての方が、当然ながら割当てられた PE 間に介在する PE の数も少なく、割当て後のデータ転送量の最大値も小さくなる可能性が高い。広い範囲で互いに離れた PE にプロセスを割当てたのでは介在する PE の数が多く、別のプロセスのデータ転送と重なる可能性が高く、従っ

50

て、割当て後のデータ転送量の最大値も大きくなる可能性が高い。

【0050】すなわち、図13のフローチャートは、プロセス割当て後に残された利用可能なPEがまとまってくるべく狭い範囲に存在するような指標を与えることになる。すなわち、プロセス割当て後に、残された利用可能なPEを次回割当ての際に、データ転送効率が高くなるべくよくなるような指標を与えていることになる。

【0051】

【発明が解決しようとする課題】前記のような従来のものにおいては、次のような課題があった。

(1)：従来例1では、プロセッサの能力はまだ十分に開いているにも関わらず、データ転送ができないため、全体の処理が実時間で実現不可能になってしまうことがある。

【0052】(2)：従来例2では、前記プロセス割当て方法は通信経路が複数あれば効果はあるが、リニアアレイ型プロセッサシステムでは、通信路は1つ、リンク構成でも2つしかなく、ほとんど効果がない。

【0053】(3)：従来例3では、制御グループを先に静的に決定してしまうため、グループ間のデータ転送量の負荷分散を行うには、その都度、制御グループの割当て変更が必要となり、負荷分散が動的に行われな

い。【0054】(4)：従来例4に示したリニアアレイ型プロセッサシステムにおける動的プロセス割当てにおいて、割当てすべきプロセスは複数のプロセッサ資源を要求する。従って、一般的に、各プロセスは単一プロセッサで処理されるサブプロセスを複数接続した構成となっている。サブプロセスの実行スケジュールによっては、そのプロセスが要求するプロセッサ数や、データ転送量は可変である。

【0055】従って、このサブプロセスの実行スケジュールが動的プロセス割当てによるプロセッサ間データ転送量の増減に影響し、このことで、システム全体の能力を十分に発揮することができなくなってしまうという課題がある。

【0056】また、プロセスが要求するプロセッサ数やデータ転送量の少ないサブプロセスの実行スケジュールを使うことで、他のプロセスの実行速度は満足できても、そのプロセス単体の処理速度が低下してしまい、全体の処理が実現できなくなるという課題がある。

【0057】本発明は、このような従来の課題を解決し、リニアアレイ型プロセッサシステムにおける動的プロセス割当てによってシステム全体の能力を最大限引き出すために、サブプロセスの実行スケジュールを、予め静的スケジューリング手段によって求めておくことで、効率的な処理を可能にすることを目的とする。

【0058】

【課題を解決するための手段】本発明は前記の目的を達成するため、次のように構成した。

(1)：複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるプロセス割当て方法において、前記動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、サブプロセスの静的スケジューリング手段により静的に求めておくようにした。

【0059】(2)：前記(1)のプロセス割当て方法において、前記サブプロセスの静的スケジューリング手段が、そのプロセスが要求するプロセッサ数最小となるスケジュール情報を得るようにした。

【0060】(3)：前記(1)のプロセス割当て方法において、前記サブプロセスの静的スケジューリング手段が、そのプロセス単体の処理時間最小となるスケジュール情報を得るようにした。

【0061】(4)：前記(1)のプロセス割当て方法において、前記サブプロセスの静的スケジューリング手段が、そのプロセスが要求するプロセッサ間データ転送量最小となるスケジュール情報を得るようにした。

【0062】(5)：前記(1)のプロセス割当て方法において、前記サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュール情報を選択して割当てるようにした。

【0063】(6)：複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるためのサブプロセスのスケジュール情報を、予め、静的に求めておくサブプロセスの静的スケジューリング手段を備えていることを特徴とした静的スケジューリング装置。

【0064】(7)：コンピュータに、複数のプロセッサを直列接続したリニアアレイ型マルチプロセッサシステムの前記各プロセッサに対し、複数のサブプロセスを構成要素として含むプロセスを動的に割当てるためのサブプロセスのスケジュール情報を、予め、静的に求めておく手順を実行させるためのプログラムを記録したコンピュータ読取可能な記録媒体。

【0065】(作用) 前記構成に基づく本発明の作用を説明する。

(a)：前記(1)の作用

サブプロセスの静的スケジューリング手段は、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセスを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

【0066】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄

11

与するところが大である。

【0067】(b)：前記(2)の作用

サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセスが要求するプロセッサ数最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、そのプロセスが要求するプロセッサ数が最小となるサブプロセススケジュールを求めることで、プロセッサ数の少ないシステムに適した動的プロセス割当てを行うことができる。

【0068】(c)：前記(3)の作用

サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセス単体の処理時間最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、そのプロセス単体の処理時間が最小となるサブプロセススケジュールを求めることで、全プロセスの処理時間の短い動的プロセス割当てを行うことができる。

【0069】(d)：前記(4)の作用

サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセスが要求するプロセッサ間データ転送量最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、そのプロセスが要求するプロセッサ間データ転送量の最大値が最小となるサブプロセススケジュールを求めることで、プロセッサ間データ転送のための、容量が小さなシステムに適した動的プロセス割当てを行うことができる。

【0070】(e)：前記(5)の作用

サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、前記サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュール情報を選択して割当てる。

【0071】このようにすれば、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、複数のサブプロセススケジュールを求めておき、その中から動的プロセス割当て時に、最適なサブプロセススケジュールを選択することで、その時の割当て状況に適した動的プロセス割当てを行うことができる。

【0072】(f)：前記(6)の作用

静的スケジューリング装置のサブプロセスの静的スケジューリング手段は、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセスを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

12

【0073】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄与するところが大である。

【0074】(g)：前記(7)の作用

コンピュータが、前記記録媒体のプログラムを読み出して実行することにより、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセスを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

【0075】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄与するところが大である。

【0076】

20 【発明の実施の形態】以下、発明の実施の形態を図面に基づいて詳細に説明する。

§ 1：システムの説明・・・図 1 参照

図 1 はシステムの説明図である。以下、図 1 に基づいて本実施の形態で使用するシステムについて説明する。図 1 に示したように、このシステムは、リニアアレイ型マルチプロセッサシステムに対し、動的プロセス割当てを行うシステムである。

30 【0077】前記リニアアレイ型マルチプロセッサシステムは、複数のプロセッサ 3 2 (PE 1、PE 2、PE 3、PE 4・・・) が直列接続されたシステムであり、前記従来例 4 と同じ構成のシステムである。この場合、前記プロセッサ 3 2 は、それぞれ、前記従来例 4 の情報処理ユニット 1 0 に対応している。

40 【0078】そして、各プロセッサ 3 2 に対して、動的プロセス割当てを行うホストプロセッサ 3 1 を前記各プロセッサ 3 2 とバスにより接続されている。また、このホストプロセッサ 3 1 により、前記各プロセッサ 3 2 に動的プロセス割当てを行う前に、サブプロセスの静的割当てを行うための静的スケジューリング装置 3 3 を前記ホストプロセッサ 3 1 に接続しておく。

【0079】ホストプロセッサ 3 1 には、各プロセッサ 3 2 に対して動的プロセス割当てを行うための動的プロセス割当て手段 3 4 が設けてあり、静的スケジューリング装置 3 3 には、サブプロセスの静的スケジューリングを行うためのサブプロセスの静的スケジューリング手段 3 5 が設けてある。

50 【0080】この場合、静的スケジューリング装置 3 3 は、リニアアレイ型プロセッサシステムに対し、バス結合等により直接接続しても良いが、このような接続でなくとも良い。例えば、静的スケジューリング装置 3 3



13

と、ホストプロセッサ 3 1 の間は、何らかの手段によりデータ通信が可能な状態にしておけば良く、LAN等の通信回線を介して接続し、互いにデータ伝送可能な状態で使用することも可能である。

【0081】すなわち、プロセッサ 3 3 は、前記ホストプロセッサ 3 1 の近く（例えば、同一パーソナルコンピュータ内）に置いても良いし、離れた場所（異なる装置内）に置いても良い。少なくとも、動的プロセス割当て手段 3 4 を備えた装置へデータ伝送できれば実施可能である。この場合、前記動的プロセス割当て手段 3 4、及びサブプロセスの静的スケジューリング手段 3 5 は、プログラムの実行により実現するものである。

【0082】§ 2：プロセス割当て方法の説明・・・図 2 参照

図 2 はサブプロセスの処理説明図である。以下、図 2 に基づいて、サブプロセスの処理を説明する。図 1 に示したシステムにおいて、ホストプロセッサ 3 1 が各プロセッサ 3 2 に対し、動的プロセス割当て処理を行うことで、前記リニアアレイ型プロセッサシステムによる処理を行う。

【0083】この場合、割当てる各プロセスは、一般に複数のプロセッサ資源を要求するため、単一プロセッサで処理されるサブプロセスを複数接続した構成になっている。このサブプロセスの実行によって、そのプロセスが要求する資源は変化する。以下の説明では、前記サブプロセスは、1つのプロセスの要素を構成しており、1つのプロセスに複数のサブプロセスが存在する場合について説明する。

【0084】前記のように、ホストプロセッサ 3 1 が動的プロセス割当てを行う時、前記動的割当てを行うプロセスを構成する複数のサブプロセスを、動的プロセス割当てを行う前に、静的スケジューリング装置 3 3 に設けたサブプロセスの静的スケジューリング手段 3 5 が、静的にスケジューリングを行う。

【0085】そこで求められたサブプロセスの実行スケジュール情報から、そのプロセスの要求するプロセッサ数や、プロセッサ間データ転送量を、前記ホストプロセッサ 3 1 へ転送し、動的プロセス割当て手段 3 4 が受信する。その後、ホストプロセッサ 3 1 の動的プロセス割当て手段 3 4 は、受信した前記プロセスの要求するプロセッサ数や、プロセッサ間データ転送量を基に、各プロセッサ 3 2 に対して動的プロセス割当てを行う。このようにして、全体の処理の効率の良い実現を可能にしている。

【0086】この場合、前記リニアアレイ型マルチプロセッサシステムの各プロセッサ 3 2 に対し、サブプロセスを構成要素として含むプロセスの割当てを行うプロセス割当て方法には、次のような方法がある。

【0087】①：プロセス割当て方法 1 は、前記サブプロセスの静的スケジューリング手段 3 5 が、そのプロセ

14

スが要求するプロセッサ数最小となるスケジュールを得る方法（時間制約付スケジューリング）である。

【0088】②：プロセス割当て方法 2 は、サブプロセスの静的スケジューリング手段 3 5 が、そのプロセス単体の処理時間が最小となるスケジュールを得る方法（プロセッサ数制約付スケジューリング）である。

【0089】③：プロセス割当て方法 3 は、サブプロセスの静的スケジューリング手段 3 5 が、そのプロセスが要求するプロセッサ間データ転送量が最小となるスケジュールを得る方法である。

【0090】④：プロセス割当て方法 4 は、前記サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュールを選択して割当てる方法である。

【0091】§ 3：プロセス割当て方法 1 の説明・・・図 3 参照

図 3 はプロセス割当て処理フローチャート（その 1）である。プロセス割当て方法 1 は、サブプロセスの静的スケジューリング手段 3 5 が、そのプロセスが要求するプロセッサ数が最小となるスケジュールを得る方法（時間制約付スケジューリング）であり、以下、図 3 に基づいて説明する。なお、以下の処理はサブプロセスの静的スケジューリング手段 3 5 が行う処理であり、S 5 1～S 5 4 は各処理ステップを示す。

【0092】サブプロセスの静的スケジューリング手段 3 5 は静的スケジューリングを行う時、指定した時間内にそのプロセスが終了するために要求されるプロセッサ数  $N_p$  が最小となるサブプロセススケジュールを求める。従って、先ず、前記プロセッサ数  $N_p$  を、 $N_p =$ （サブプロセスの処理時間の総和）／（目的の処理時間）の式により求める（S 5 1）。

【0093】次に、 $N_p$  プロセッサで割当てを試み（S 5 2）、割当てが可能か否かを判断する（S 5 3）。その結果、割当てが可能であれば処理を終了するが、割当てが不可能であれば、 $N_p$  を更新（ $N_p = N_p + 1$ ）し（S 5 4）、前記 S 5 2 の処理から繰り返して行う。このようにして、前記 S 5 3 の処理で割当てが可能になった時点で、そのプロセスが要求するプロセッサ数が最小となるスケジュールを得ることができる。

【0094】§ 4：プロセス割当て方法 2 の説明・・・図 4 参照

図 4 はプロセス割当て処理フローチャート（その 2）である。プロセス割当て方法 2 は、サブプロセスの静的スケジューリング手段 3 5 が、そのプロセス単体の処理時間が最小となるスケジュールを得る方法（プロセッサ数制約付スケジューリング）であり、以下、図 4 に基づいて説明する。なお、以下の処理はサブプロセスの静的スケジューリング手段 3 5 が行う処理であり、S 6 1～S 6 4 は各処理ステップを示す。

【0095】サブプロセスの静的スケジューリング手段

10

20

30

40

50



15

35は静的スケジューリングを行う時、指定したプロセッサ数でそのプロセスが終了するための処理時間Tが最小となるサブプロセススケジュールを求める。従って、先ず、前記処理時間Tを、 $T = (\text{サブプロセスの処理時間の総和}) / (\text{目的のプロセッサ数})$ の式により求める(S61)。

【0096】次に、処理時間 $\leq T$ の条件で割当てを試み(S62)、割当てが可能か否かを判断する(S63)。その結果、割当てが可能であれば処理を終了するが、割当てが不可能であれば、Tを更新  $T = T + (\text{単位時間})$  し(S64)、前記S62の処理から繰り返して行う。このようにして、前記S63の処理で割当てが可能になった時点で、そのプロセス単体の処理時間が最小となるスケジュールを得ることができる。

【0097】§5：プロセス割当て方法3の説明・・・  
図5参照

図5はプロセス割当て処理フローチャート(その3)である。また、図6はプロセス割当て処理説明図(その1)である。プロセス割当て方法3は、サブプロセスの静的スケジューリング手段35が、そのプロセスが要求するプロセッサ間データ転送量が最小となるスケジュールを得る方法であり、以下、図5、図6に基づいて説明する。なお、以下の処理はサブプロセスの静的スケジューリング手段35が行う処理であり、S71～S79は各処理ステップを示す。また、Nはサブプロセスの数、Miはi分割の分割パターン数である。

【0098】サブプロセスの静的スケジューリング手段35は、先ず、プロセスを、サブプロセスの数をNとした時、 $i = 1$  からNまで分割を試みる。すなわち、 $i = 1 \sim N$ ループとして(S71)、プロセスを分割する(S72)。この場合、i分割する時の分割パターンと

の数をMiとすると、分割パターンは、 $j = 1$  からMiループまで存在する(S73)。  
【0099】例えば、 $N = 3$ で、 $\{1, 2, 3\}$ を2分割する場合、 $\{1, 2\}$ 、 $\{3\}$ 、 $\{1\}$ 、 $\{2, 3\}$ のM2=2パターン存在する。従って、 $j = 1 \sim Mi$ ループとして(S73)、分割されたプロセスに対してスケジューリングを行い、各サブプロセスの実行開始時刻を求める(S74)。この時、条件(時間制約)を満たすことができるか否かを判断し(S75)、前記条件を満たせばS76へ進み、満たさなければ、S78へ進み、別の分割が試される。

【0100】前記条件を満たした場合(時間制約を満たす場合)、その時の各分割間のデータ転送量の最大値が最小となるか否かを判断し(S76)、最小とならなければS78の処理を行い、最小となれば、データ転送量の最大値が最小となるものをメモリ等に保存する(S77)。

【0101】これを繰り返し実行し、全ての分割パターンに対して行う。そして、2重ループ(S78、S7

16

9)が終了した時、前記メモリ等に保存されているスケジュールをそのプロセスのサブプロセススケジュールとする。

【0102】以下、図6に基づいて、前記プロセス割当て方法3の処理を具体的に説明する。図6に示したプロセスは、8個のサブプロセス  $\{1, 2, 3, 4, 5, 6, 7, 8\}$  から構成される。これを  $i = 3$  の時、すなわち、3分割の場合、例えば、 $\{1, 5, 7\}$ 、 $\{2, 4\}$ 、 $\{3, 6, 8\}$  のように分割される。

【0103】この結果に対してスケジューリングを行い、3プロセッサに対して、各サブプロセスの開始時刻を決定する。そして、条件(時間制約)を満たすスケジュールが存在すれば、転送量を計算し、その内の最大値が、以前に求めた割当てより小さい場合は、これを保存する。

【0104】§6：プロセス割当て方法4の説明・・・  
図7参照

図7はプロセス割当て処理説明図(その2)である。プロセス割当て方法4は、サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュールを選択して割当てる方法であり、以下、図7に基づいて説明する。

【0105】この処理では、前記サブプロセスの静的スケジューリング手段35がN個で構成されており、これらのN個の手段によりサブプロセスの静的スケジューリングを行う。前記静的スケジューリングを行う時、前記N個のサブプロセスの静的スケジューリング手段(1～N)により、予め求めておいた複数のサブプロセススケジュールから、動的プロセス割当て時に最適なサブプロセススケジュール結果を選択して使用する。

【0106】1. 現在のプロセッサへのプロセス割当て状況(どのプロセッサが空いているか)

2. 現在のプロセッサ間のデータ転送量

3. 将来のプロセス割当て解除情報(いつ、どのプロセッサから、どのプロセスが割当て解除されるか)

前記処理において、前記1～3に示した3つの情報をプロセスに割当てる毎に更新している。次のプロセスを割当てる時、この3つの情報から例えば、プロセッサの空きが少ない場合は、要求プロセッサ数が最小となるサブプロセッサ割当てを選択してプロセス割当てを行う。また、データ転送量の空きが少ない時には、データ転送量が最小となるサブプロセッサ割当てを選択してプロセス割当てを行う。

【0107】更に、次に行うプロセスの処理時間が最小となるサブプロセス割当ては、この処理時間と割当て解除になる時間との和が、このプロセスの時間制約以内に納まる時使用される。

【0108】§7：記録媒体とプログラムの説明

前記静的スケジューリング装置33のサブプロセスの静的スケジューリング手段35が行うサブプロセスの静的

17

スケジューリング処理は、静的スケジューリング装置 33 内の CPU がプログラムを実行することにより、次のようにして実現する。

【0109】静的スケジューリング装置 33 にはハードディスク装置が設けてあり、このハードディスク装置の記録媒体（ハードディスク）に、前記処理を実現するためのプログラムやその他の各種データ等を格納しておく。そして、前記処理を行う場合は、CPU の制御によりハードディスク装置の記録媒体に格納されている前記プログラムやデータを読み出して静的スケジューリング装置 33 内に設けたメモリに取り込む。

【0110】その後、CPU が前記メモリに格納してあるプログラムの内、必要なプログラムから順次読み出して実行することにより、前記処理を行う。なお、前記ハードディスク装置の記録媒体に記録するプログラムは、次のようにして記録（記憶）する。

【0111】①：フレキシブルディスク（フロッピーディスク）に格納されているプログラム（他の装置で作成したプログラムデータ）を、静的スケジューリング装置 33 に設けたフレキシブルディスクドライブ装置により読み取り、ハードディスク装置の記録媒体（ハードディスク）に格納する。

【0112】②：光磁気ディスク、或いは CD-ROM 等の記憶媒体に格納されているデータを、静的スケジューリング装置 33 に設けたドライブ装置により読み取り、ハードディスク装置の記録媒体（ハードディスク）に格納する。

【0113】③：LAN 等の通信回線を介して他の装置から伝送されたデータを静的スケジューリング装置 33 で受信し、そのデータをハードディスク装置の記録媒体（ハードディスク）に格納する。

【0114】

【発明の効果】以上説明したように、本発明によれば次のような効果がある。(1)：請求項 1 では、サブプロセスの静的スケジューリング手段は、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセスを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

【0115】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄与するところが大である。

【0116】(2)：請求項 2 では、サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセスが要求するプロセッサ数最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジュー

18

リングを行う時、そのプロセスが要求するプロセッサ数が最小となるサブプロセススケジュールを求めることで、プロセッサ数の少ないシステムに適した動的プロセス割当てを行うことができる。

【0117】(3)：請求項 3 では、サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセス単体の処理時間最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、そのプロセス単体の処理時間が最小となるサブプロセススケジュールを求めることで、全プロセスの処理時間の短い動的プロセス割当てを行うことができる。

【0118】(4)：請求項 4 では、サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、そのプロセスが要求するプロセッサ間データ転送量最小となるスケジュール情報を得る。このように、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、そのプロセスが要求するプロセッサ間データ転送量の最大値が最小となるサブプロセススケジュールを求めることで、プロセッサ間データ転送のための、容量が小さなシステムに適した動的プロセス割当てを行うことができる。

【0119】(5)：請求項 5 では、サブプロセスの静的スケジューリング手段は、サブプロセスのスケジュール情報を求める時、前記サブプロセスのスケジュール情報を、複数の条件により求めておき、プロセスの要求条件に応じて適当なスケジュール情報を選択して割当てる。

【0120】このようにすれば、サブプロセスの静的スケジューリング手段が静的スケジューリングを行う時、複数のサブプロセススケジュールを求めておき、その中から動的プロセス割当て時に、最適なサブプロセススケジュールを選択することで、その時の割当て状況に適した動的プロセス割当てを行うことができる。

【0121】(6)：請求項 6 では、静的スケジューリング装置のサブプロセスの静的スケジューリング手段は、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセスを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

【0122】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄与するところが大である。

【0123】(7)：請求項 7 では、コンピュータが、前記記録媒体のプログラムを読み出して実行することにより、動的プロセス割当てを行うためのサブプロセスのスケジュール情報を、予め、静的に求めておく。このように、動的プロセス割当てを行う時、割当てを行うプロセ

19

スを構成するサブプロセスを動的プロセス割当てを行う前に、予め静的にスケジュール情報を求めておく。

【0124】そして、そこで求められたサブプロセスの実行スケジュールからそのプロセスの要求するプロセッサ数やプロセッサ間データ転送量を基に、動的プロセス割当てを行うことで、全体の処理の効率の良い実現に寄与するところが大である。

#### 【図面の簡単な説明】

【図1】本発明の実施の形態におけるシステムの説明図である。

【図2】本発明の実施の形態におけるサブプロセスの処理説明図である。

【図3】本発明の実施の形態におけるプロセス割当て処理フローチャート（その1）である。

【図4】本発明の実施の形態におけるプロセス割当て処理フローチャート（その2）である。

【図5】本発明の実施の形態におけるプロセス割当て処理フローチャート（その3）である。

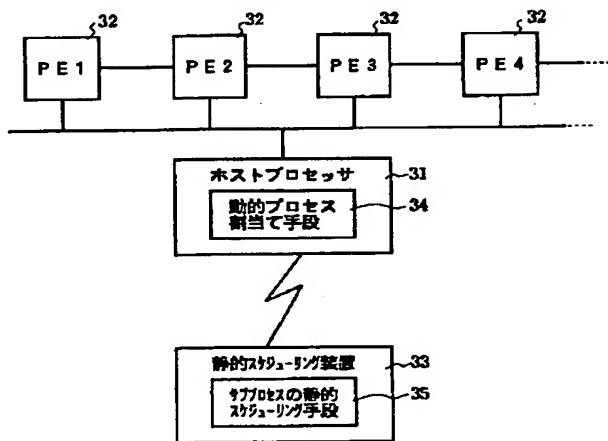
【図6】本発明の実施の形態におけるプロセス割当て処理説明図（その1）である。

【図7】本発明の実施の形態におけるプロセス割当て処理説明図（その2）である。

【図8】従来例の説明図（その1）である。

【図1】

システムの説明図



20

\* 【図9】従来例の説明図（その2）である。

【図10】従来例の説明図（その3）である。

【図11】従来例の説明図（その4）である。

【図12】従来例の説明図（その5）である。

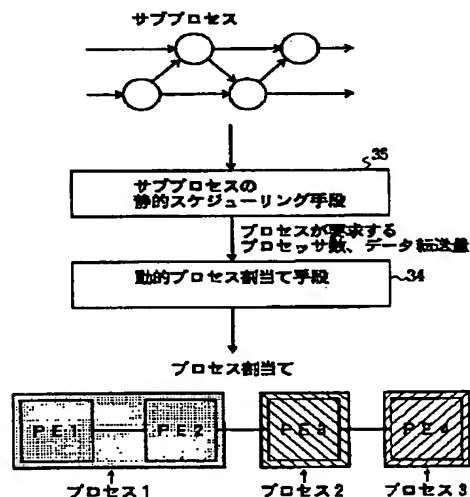
【図13】従来例の説明図（その6）である。

#### 【符号の説明】

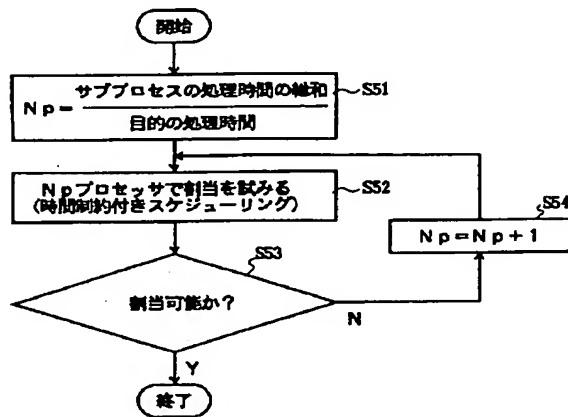
- 10 情報処理ユニット
- 11 信号処理プロセッサ
- 12 命令キャッシュ
- 13 データRAM
- 14、15 リンク制御部
- 16 メインキャッシュ
- 17 リンクキャッシュ
- 18 DRAM
- 19 DRAMコントローラ
- 20 通信リンク
- 25 信号処理部
- 26 通信制御部
- 30 ホストメモリバス
- 31 ホストプロセッサ
- 33 静的スケジューリング装置
- 34 動的プロセス割当て手段
- 35 サブプロセスの静的スケジューリング手段

【図2】

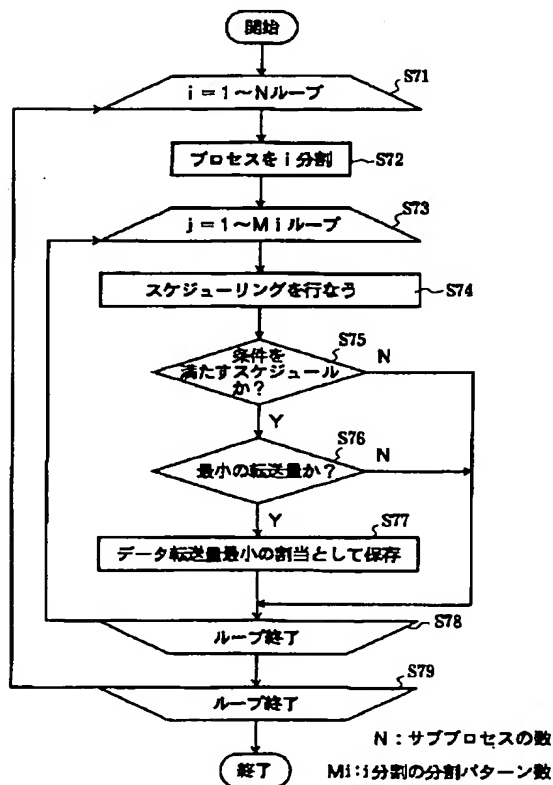
サブプロセスの処理説明図



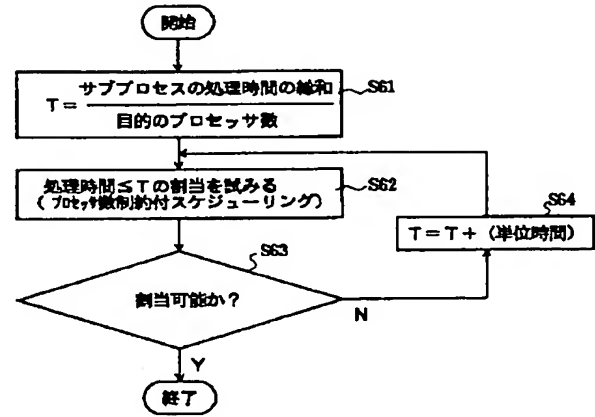
【図 3】

プロセス割当て処理フローチャート  
(その 1)

【図 5】

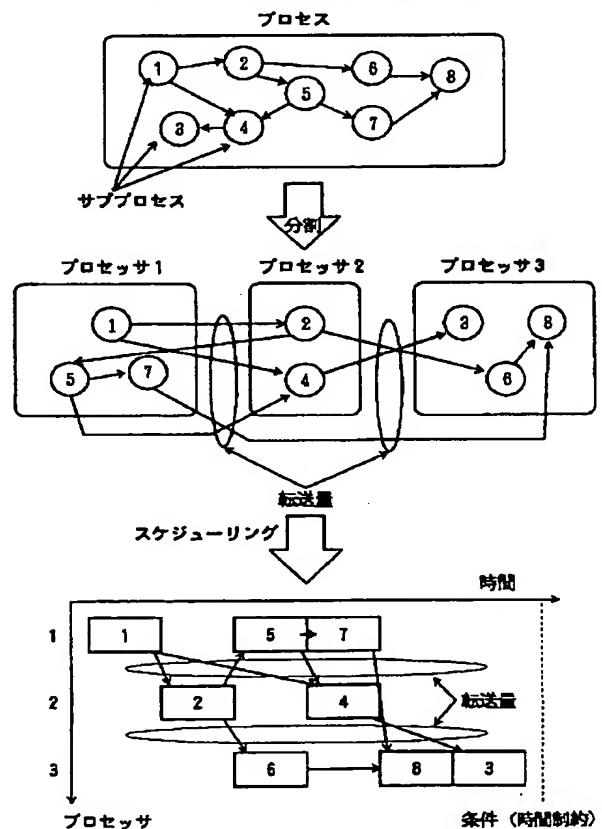
プロセス割当て処理フローチャート  
(その 3)

【図 4】

プロセス割当て処理フローチャート  
(その 2)

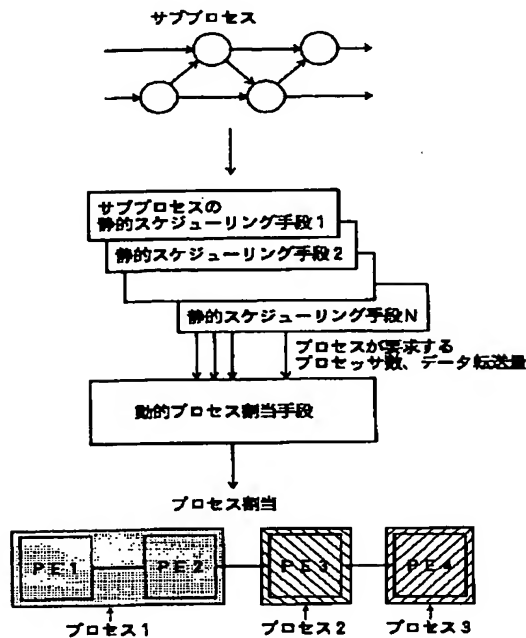
【図 6】

## プロセス割当て処理説明図 (その 1)



【図 7】

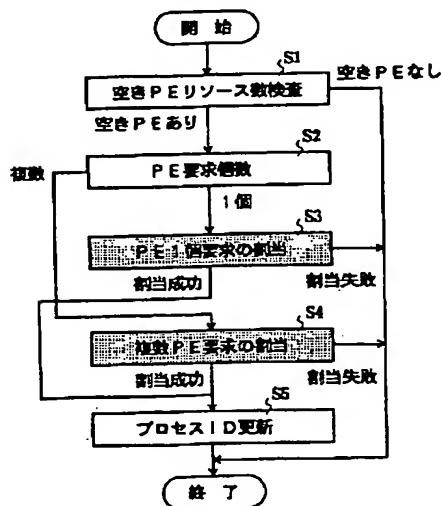
## プロセス割当て処理説明図 (その 2)



【図 10】

## 従来例の説明図 (その 3)

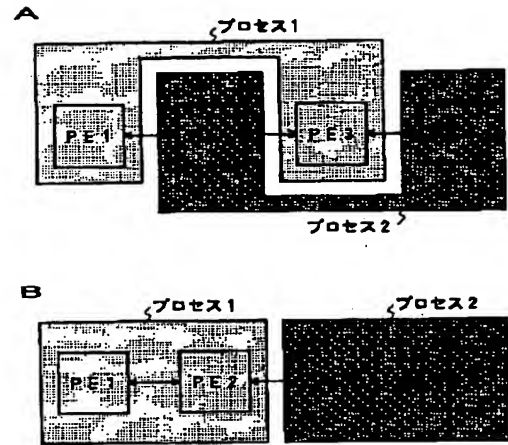
動的プロセス割当てアルゴリズム  
のメインルーチンを示すフローチャート



【図 9】

## 従来例の説明図 (その 2)

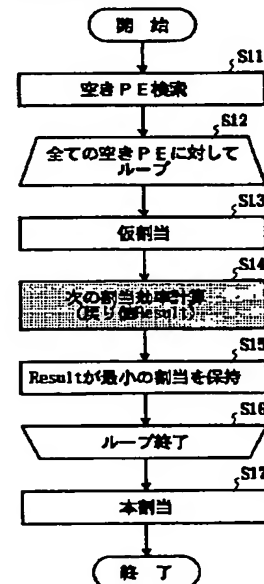
A 及び B は、2 つの異なるプロセス割当ての様子を示す図



【図 11】

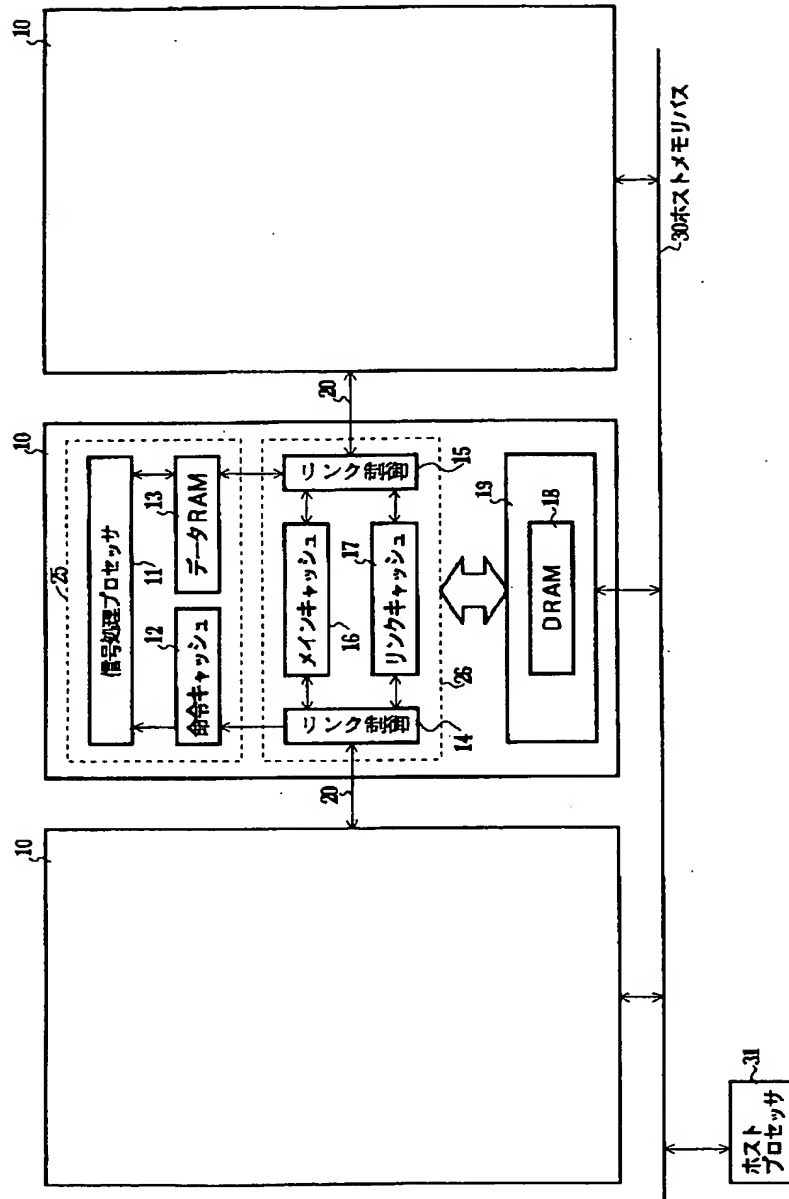
## 従来例の説明図 (その 4)

図 10 のステップ S3 に於ける PE の 1 個要求  
に対する割当て処理を示すフローチャート



【図 8】

## 従来例の説明図（その 1）

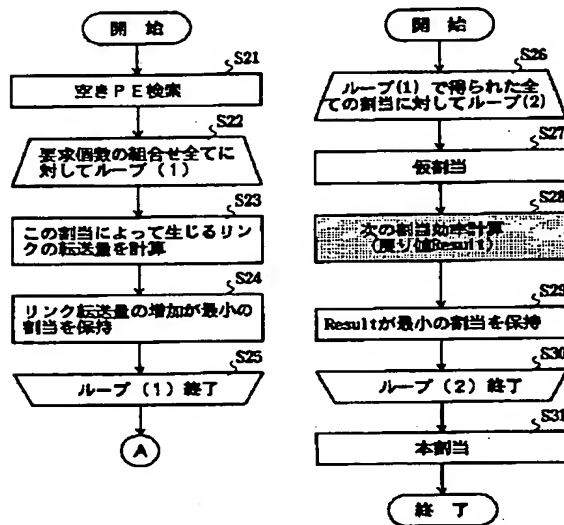




【図 12】

## 従来例の説明図（その 5）

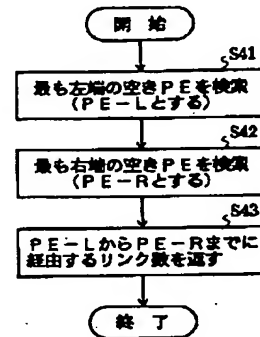
図 10 のステップ S4 に於ける PE の複数個要求に対する割当てを示すフローチャート



【図 13】

## 従来例の説明図（その 6）

図 11 のステップ 14 及び図 12 のステップ 28 に於ける次の割当て処理計算処理を示すフローチャート



**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**